

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re U.S. Patent Application of )  
KAMEYAMA et al. )  
Application Number: To Be Assigned )  
Filed: Concurrently Herewith )  
For: DATA TRANSFER METHOD )

#3  
JC971 U S PRO  
09/940506  
08/29/01  


Honorable Assistant Commissioner  
for Patents  
Washington, D.C. 20231

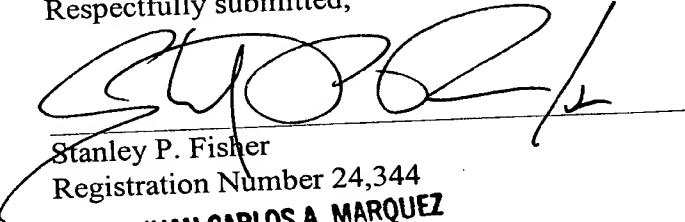
**REQUEST FOR PRIORITY  
UNDER 35 U.S.C. § 119  
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of November 27, 2000, the filing date of the corresponding Japanese patent application 2000-364542.

The certified copy of corresponding Japanese patent application 2000-364542 is being submitted herewith. Acknowledgment of receipt of the certified copies is respectfully requested in due course.

Respectfully submitted,



Stanley P. Fisher

Registration Number 24,344

JUAN CARLOS A. MARQUEZ  
Registration No. 34,072

**REED SMITH HAZEL & THOMAS LLP**  
3110 Fairview Park Drive  
Suite 1400  
Falls Church, Virginia 22042  
(703) 641-4200  
August 29, 2001



日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日  
Date of Application:

2000年11月27日

出願番号  
Application Number:

特願2000-364542

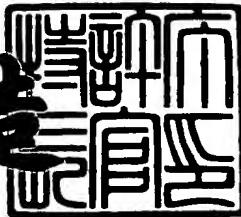
出願人  
Applicant(s):

株式会社日立製作所

2001年 4月20日

特許庁長官  
Commissioner,  
Japan Patent Office

及川耕造



出証番号 出証特2001-3033961

【書類名】 特許願  
 【整理番号】 H00013151A  
 【提出日】 平成12年11月27日  
 【あて先】 特許庁長官 殿  
 【国際特許分類】 G06F 11/20  
 【発明者】  
   【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地  
     株式会社日立製作所中央研究所内  
   【氏名】 亀山 伸  
 【発明者】  
   【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地  
     株式会社日立製作所中央研究所内  
   【氏名】 垂井 俊明  
 【発明者】  
   【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地  
     株式会社日立製作所中央研究所内  
   【氏名】 今木 常之  
 【発明者】  
   【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目280番地  
     株式会社日立製作所中央研究所内  
   【氏名】 川本 真一  
 【特許出願人】  
   【識別番号】 000005108  
   【氏名又は名称】 株式会社日立製作所  
 【代理人】  
   【識別番号】 100075096  
   【弁理士】  
   【氏名又は名称】 作田 康夫  
   【電話番号】 03-3212-1111

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 通信処理方法ならびに通信処理プログラムが記録される記録媒体

【特許請求の範囲】

【請求項1】

第1の情報処理装置上で実行されるプロセスと第2の情報処理装置上で実行されるプロセスが互いのユーザ空間同士の直接転送により通信を行う機能を有する情報処理システムにおいて、

前記第1の情報処理装置上の第1のプロセスと前記第2の情報処理装置上の第2のプロセスとの間の第1の接続による通信を、前記第1の情報処理装置上の第3のプロセスと前記第2のプロセスとの間の第2の接続によって引き継いで通信を継続する通信処理方法であって、

前記第1のプロセスと前記第2のプロセスが前記第1の接続による通信を中断する第1のステップと、

前記第1のプロセスの要求により前記第3のプロセスと前記第2のプロセスが新たに第2の接続を確立する第2のステップと、

前記第1のプロセスの要求により前記第2の接続が前記第1の接続による通信を引き継いで通信を継続する第3のステップとを有することを特徴とする通信処理方法。

【請求項2】

前記第3のプロセスがプロセス生成機能により生成されることを特徴とする請求項1記載の通信処理方法。

【請求項3】

前記第1のステップは

前記第1のプロセスが前記第2のプロセスに対してデータ転送の中断要求を行うこと、

第2のプロセスが前記中斷要求に応じて前記前記第1のプロセスに対するデータ転送を中斷し、中斷完了後に前記第1のプロセスに対して中斷完了報告を行うこと、

との手順を有することを特徴とする請求項1記載の通信処理方法。

【請求項4】

前記第1のプロセスにて、前記第2のプロセスに中断要求を行ってからデータ受信処理があったこと記憶し、前記中断完了報告を受けた時点以降該データ受信処理で受信したデータを前記第3のプロセスにコピーすることを特徴とする請求項3の通信処理方法。

【請求項5】

前記第2のステップは、  
前記第2の接続が確立されたことを前記第2のプロセスと前記第3のプロセスのいずれか一方、または両方が前記第1のプロセスに報告する手順を含むことを特徴とする請求項1記載の通信処理方法。

【請求項6】

前記第2のステップにおける前記第2の接続の確立の要求は、前記第1のプロセスにて自発的事象が発生したことを契機に発せられることを特徴とする請求項1記載の通信処理方法。

【請求項7】

前記第2のステップにおける前記第2の接続の確率の要求は、前記第1のプロセスにて突発的事象が発生したことを契機に発せられることを特徴とする請求項1記載の通信処理方法。

【請求項8】

前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデータ受信処理が発生したことを検出し記憶することを特徴とする請求項1記載の通信処理方法。

【請求項9】

前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデータ受信処理が発生したことを検出した場合、前記第3のプロセスに対して通知することを特徴とする請求項7記載の通信処理方法。

【請求項10】

前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデ

ータ受信処理が発生したことを検出し、前記第3のプロセスに対して通知した場合、前記第1のプロセスと前記第3のプロセスが強調して該データを前記第1のプロセスから前記第3のプロセスにコピーすることを特徴とする請求項7記載の通信処理方法。

【請求項11】

前記第2の接続の確立の完了後に前記第1の接続を切断することを特徴とする請求項1記載の通信処理方法。

【請求項12】

ネットワークで相互に接続された第1の情報処理装置と第2の情報処理装置とを有し、該第1の情報処理装置上で実行されるプロセスと第2の情報処理装置上で実行されるプロセスが互いのユーザ空間同士の直接転送により通信を行う機能を有する情報処理システムにおいて、

前記第1のプロセスと前記第2のプロセスが前記第1の接続による通信を中断する第1の手段と、

前記第1のプロセスの要求により前記第3のプロセスと前記第2のプロセスが新たに第2の接続を確立する第2の手段と、

前記第1のプロセスの要求により前記第2の接続が前記第1の接続による通信を引き継いで通信を継続する第3の手段と有することを特徴とする情報処理システム。

【請求項13】

請求項1記載の通信処理方法がソケット通信の機能をエミュレートするようにプログラムされたエミュレーションライブラリで実現され、前記第1の情報処理装置と前記第2の情報処理装置でそれぞれ前記エミュレーションライブラリを実行することにより、前記第1の情報処理装置と前記第2の情報処理装置のそれぞれで実行するソケット通信のためのユーザプログラムを何ら変更することなく通信を可能とした請求項1記載の通信処理方法。

【請求項14】

第1の情報処理装置上の第1のプロセスと前記第2の情報処理装置上の第2のプロセスとの間の第1の接続による通信を、前記第1の情報処理装置上の第3の

プロセスと前記第2のプロセスとの間の第2の接続によって引き継いで通信を継続する通信処理方法であって、

前記第1のプロセスと前記第2のプロセスが前記第1の接続による通信を中断する第1のステップと、

前記第1のプロセスの要求により前記第3のプロセスと前記第2のプロセスが新たに第2の接続を確立する第2のステップと、

前記第1のプロセスの要求により前記第2の接続が前記第1の接続による通信を引き継いで通信を継続する第3のステップとを有することを通信処理方法を実行するエミュレーションライブラリが記録された計算機利用可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、TCP/IPのソケット通信におけるforkをエミュレートする通信処理方法に係り、特にOSの介在なしにユーザ空間転送を行う情報処理システムにおいて、forkで生成された子プロセスが親プロセスとクライアント間のネットワーク接続を引き継ぐ通信処理方法に関する。

【0002】

【従来の技術】

サーバ・クライアントモデルの分散型情報処理システムにおいて、TCP/IPによるソケット通信はごく一般的であり、莫大なソフトウェア資産が存在する。図2にサーバ・クライアントモデルの情報処理システムの例を示す。この例ではホスト210とホスト260が通信回線30で結合しており、サーバアプリケーション211とクライアントアプリケーション261がソケット通信を行う。

【0003】

従来のソケット通信では、クライアントアプリケーション261のプロセス262からの要求をサーバアプリケーション211の親プロセス212が受けた場合、親プロセス212がUNIXのfork機能によって子プロセス213を生成し、子プロセス213がプロセス262からの要求を処理する。しかしながらソケット通信では通信データを一旦OS空間内のバッファ（バッファ231やバッ

ファ281)でバッファリングするために、スループットの向上に限界があった。

#### 【0004】

これに対して近年、ユーザ空間で直接データ転送を行いOS空間へのコピーを不要とするVIA (Virtual Interface Architecture) 等の次世代高速IO方式が提案されている。公知の技術としては例えばSteven H. Rodrigues他による“High-Performance Local Area Communication With Fast Socket” (Proceedings of the USENIX、1997) がある。該技術によると高速ネットワークでサーバとクライアント間を接続することで性能向上を実現することが可能になり、かつソケット通信の一部の機能を該高速ネットワークのAPIでエミュレートすることでこれまでのソフトウェア資産を有効活用することが可能になる。このエミュレートに関してはUNIXのリネーム機能によって本来の関数を新しい関数で置き換えて実現できることは周知の技術である。

#### 【0005】

##### 【発明が解決しようとする課題】

ところで図2に示すように、ソケット通信における端点の一方はサーバ側（ホスト210）のプロセスが保持し（ソケット221）、他方はクライアント側（ホスト260）のプロセスが保持する（ソケット271）。

#### 【0006】

またサーバ側ではforkによって子プロセス213を生成する際に、親プロセスが保持するソケットの属性も子プロセスにコピーするため、親プロセスと子プロセスで同一のソケット221を共有して利用することが可能である。すなわちforkによって生成された子プロセス213はそのままでクライアントとの通信が可能である。

#### 【0007】

しかしながら図3に示すように、VIAではプロセス間の通信回線の端点（Virtual Interface：以後VIと記す）はそれぞれのプロセスのロ

一カルな資源であり、`fork`で子プロセス313を生成しても親プロセス312のVI331は共有できないという制限がある。

#### 【0008】

そのため`fork`によって子プロセスを生成してもクライアントとのVIの接続が確立できないのでクライアントとの通信ができないという問題が生じる。したがって、VIAでソケット通信の`fork`をエミュレートするためには生成された子プロセス313とホスト360のクライアントアプリケーション361との間でVIの接続を確立することが課題になる。

#### 【0009】

またアプリケーションによっては子プロセス313とプロセス362間でVIの接続が確立する前に親プロセス312とプロセス362間で通信が発生する場合もあり得る。この場合には親プロセス312とプロセス362間で発生した通信を確実に子プロセス313とプロセス362間で引き継ぐことが課題になる。

#### 【0010】

したがって、本発明の一つの目的は、FORKをエミュレートするようにサーバの親プロセスとクライアントのプロセス間のVI接続をサーバの子プロセスとクライアントのプロセス間に張り替えて、子プロセスが親プロセスの通信を継続する方法を提供することにある。

#### 【0011】

##### 【課題を解決するための手段】

本発明の一つの態様によれば、第1の情報処理装置上で実行されるプロセスと第2の情報処理装置上で実行されるプロセスが互いにユーザ空間直接転送により通信を行う機能を有する情報処理システムにおいて、前記第1の情報処理装置上の第1のプロセスと前記第2の情報処理装置上の第2のプロセスとの間の第1の接続による通信を、前記第1の情報処理装置上の第3のプロセスと前記第2のプロセスとの間の第2の接続によって引き継いで通信を継続する。その特徴は前記第1のプロセスと前記第2のプロセスが前記第1の接続による通信を中断する第1の機能と、前記第3のプロセスと前記第2のプロセスが前記第1のプロセスの要求により新たに第2の接続を確立する第2の機能と、前記第1のプロセスの要

求により前記第2の接続が前記第1の接続による通信を引き継いで通信を継続する第3の機能と、を有する点にある。

【0012】

また、前記第3のプロセスがプロセス生成機能により生成されることを特徴とする。

【0013】

更に、前記第1の機能は前記第1のプロセスが前記第2のプロセスに対してデータ転送の中断要求を行う機能と、前記第2のプロセスは前記中断要求に応じて前記第1のプロセスに対するデータ転送を中断し、中断完了後に前記第1のプロセスに対して中断完了報告を行う機能と、を具備することを特徴とする。

【0014】

また本発明の別の態様は、上記に加え、前記第2の機能は前記第2の接続が確立されたことを前記第2または前記第3のいずれか、または両方のプロセスが前記第1のプロセスに報告する機能を具備することを特徴とする。

【0015】

また、前記第2の機能は前記第1のプロセスで発生した自発的事象を契機に第2の接続の確立を要求することを特徴とする。

【0016】

また、前記第2の機能は前記第1のプロセスで発生した突発的事象を契機に第2の接続の確率を要求することを特徴とする。

【0017】

また、前記第3の機能は前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデータ受信処理が発生したことを検出し記憶する手段を具備することを特徴とする。

【0018】

また本発明の通信処理方法は、前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデータ受信処理が発生したことを検出した場合、前記第3のプロセスに対して通知する手段を具備することを特徴とする。

## 【0019】

また本発明の通信処理方法は、前記第2の接続が確立するまでに前記第1のプロセスが該プロセスにおいてデータ受信処理が発生したことを検出し、前記第3のプロセスに対して通知した場合、前記第1のプロセスと前記第3のプロセスが強調して該データを前記第1のプロセスから前記第3のプロセスにコピーする手段を具備することを特徴とする。

## 【0020】

また本発明の別の態様は、ソケット通信の機能をエミュレートするようにプログラムされたエミュレーションライブラリで実現され、前記第1の情報処理装置と前記第2の情報処理装置でそれぞれ前記エミュレーションライブラリを実行することにより、前記第1の情報処理装置と前記第2の情報処理装置のそれぞれで実行するソケット通信のためのユーザプログラムを何ら変更することなく通信が可能であることを特徴とする。

## 【0021】

## 【発明の実施の形態】

図1に本発明の実施形態のサーバ・クライアントシステムを示す。

## 【0022】

図1においてホスト110のメモリ（図示しない）上にはサーバアプリケーション111、TCP/IPソケットエミュレータモジュール120、VIAモジュール130が格納されている。またホスト160のメモリ（図示しない）上には同様にクライアントアプリケーション161、TCP/IPソケットエミュレータモジュール170、VIAモジュール180が格納されている。

## 【0023】

ホスト110とホスト160はネットワーク1で接続されており、サーバアプリケーション111とクライアントアプリケーション161は通信回線（10、20）を介してパケットの送受で通信を行う。

## 【0024】

TCP/IPソケットエミュレータモジュール120および170は、前記公知例と同様にVIAのAPIによってTCP/IPのソケット通信の機能をエミュ

レートした関数群を含み、特に図10に示した関数群に関しては、関数本来の機能に加え、本発明を実現するための機能を追加して該関数をエミュレートしている。

#### 【0025】

VIAモジュール130および180はVIAドライバ（図示せず）を含んでいる。またVIAによる通信回線の端点であるVI（131、132、181、182）を構成する。

#### 【0026】

サーバアプリケーション111において親プロセス112はFORK()をエミュレートした図10の参照符号F1示すEMU\_FORK()F1によって子プロセス115を生成する。EMU\_FORK()F1では子プロセスを生成する機能に、親プロセス112および子プロセス115がプロセス間通信を行うために共有メモリ117を生成し、共有メモリ117に張替え要求フラグP118、張替え要求フラグC119およびコピー発生フラグ140を生成する機能が追加されている。また親プロセス112内にデータ既受信フラグ113、張替え完了フラグ123および張替え要求監視スレッド114を生成する。さらに、子プロセス115内に張替え要求監視スレッド116およびコピー要求監視スレッド141を生成する。張替え要求フラグP118は親プロセス112によって書き込まれ、その書き込みを張替え要求監視スレッド116が監視する。コピー発生フラグ140は親プロセス112によって書き込まれ、その書き込みをコピー要求監視スレッド141が監視する。張替え要求フラグC119は子プロセス115によって書き込まれ、その書き込みを張替え要求監視スレッド114が監視する。

#### 【0027】

クライアントアプリケーション161においては、プロセス162がCONNECT()をエミュレートした図10の参照符号F2に示すEMU\_CONNECT()でもってサーバアプリケーション111に対して通信回線の確立を要求する。EMU\_CONNECT()F2には、接続が許可されるとプロセス162内に中断要求フラグ163と中断要求監視スレッド164を生成する機能が追加し

である。中断要求フラグ163はサーバアプリケーション111から送られてくるパケットによって書き込まれ、その書き込みを中断要求監視スレッド164が監視する。

#### 【0028】

図4、図5および図6は、本実施態様における親プロセス112、子プロセス115、およびプロセス162の処理手順をそれぞれ示すフローチャートである。以降、図を用いてVIの張替え処理の詳細を説明する。

#### 【0029】

まず、図4に示すフローチャートによって親プロセス112の処理の流れ説明する。

#### 【0030】

親プロセス112は、ステップS2ではクライアントアプリケーション161上のプロセス162からの接続要求を聴取する。次に、ステップS3ではプロセス162からの接続要求に対して接続許可を出す。次に、ステップS4ではEMU\_FORK()F1によって子プロセスを生成する。ステップS5ではVIの張替えが既に行われているかを張替え完了フラグ123を見てチェックする。張替え完了フラグ123はステップS13において書き込まれる。

#### 【0031】

既に張替えが行われている場合（張替え完了フラグ123が1の場合）はステップS14に進む。張替えが行われていない場合（張替え完了フラグ123が0の場合）はステップS6へ進む。

#### 【0032】

ステップS6では親プロセス112は、自プロセスでのVI張替え要求トリガの発生の有無をチェックする。子プロセスはステップS4で生成されているが、その時点（ステップS4）ではまだVIの張替えは行わない。なぜならばVIの張替えは多大なオーバヘッドを伴う高価な処理であり、また子プロセスを生成しても必ずしも該子プロセスがクライアントと通信を行うわけではないからである。そのため図7に示すように、実際に子プロセス115とプロセス162の通信を発生させるようなイベントをVI張替え処理開始のトリガとする。

## 【0033】

図7において、トリガT1は親プロセス112による通信回線10のCLOSEである。該トリガは自発的なトリガである。通信回線10のCLOSE後はプロセス162の通信相手は子プロセス115になる。

## 【0034】

トリガT2は親プロセス112の消滅である。該トリガは自発的な場合と突発的な場合がある。消滅後は子プロセス115が通信可能の状態でなければならない。

## 【0035】

トリガT3は子プロセス115の受信動作(Recv)の発生である。該トリガは自発的なトリガである。子プロセス115はプロセス162からの通信を期待しているので通信可能の状態でなければならない。トリガT4は子プロセス115の送信動作(Send)の発生である。該トリガは自発的なトリガである。子プロセス115がプロセス162へデータを送信するため通信可能の状態でなければならない。

## 【0036】

ステップS6では、親プロセス112は該トリガの発生がなければステップS14に進んで、サーバプログラムを継続して実行する。ステップS6において該トリガの発生がある場合は親プロセス112が実行している通常の処理を一旦休止し、VI張替え処理のルーチンであるステップS7に進む。

## 【0037】

ステップS7では親プロセス112はプロセス162に対してデータ転送の中斷を要求する。その後、ステップS8でプロセス162からのデータ転送中斷完了報告を待つ。データ転送中斷完了報告があると親プロセスはステップS9に進み、VI張替え前に親プロセス112とプロセス162の間で通信が発生し親プロセス112が既にデータを受け取っているか否かをチェックする。チェックはデータ既受信フラグ113を見て行う。データ既受信フラグ113はステップS16で書き込まれる。既にデータを受け取っている場合(データ既受信フラグ113が1の場合)はステップS10に進む。データを受け取っていない場合(デー

タ既受信フラグ113が0の場合)はステップS11に進む。

【0038】

ステップS10では、親プロセス112はコピー発生フラグ140を立てて子プロセス115にデータコピーの発生を通知する。続いて親プロセス112と子プロセス115が強調して親プロセス112のアドレス空間を子プロセス115にコピーし、親プロセス112が既に受信しているデータを子プロセス115に漏れなく引き継がせる。

【0039】

ステップS11では子プロセス115およびプロセス162に対して互いにV.Iを張替えるよう要求する。親プロセス112と子プロセス115は共有メモリ117を介して通信し、張替え要求フラグP118を立てることで子プロセス115に対してV.Iの張替えを要求する。

【0040】

親プロセス112とプロセス162は通信回線10を介して通信し、親プロセス112がプロセス162に対してV.I張替え要求用パケットを送信し張替え要求フラグ163を立てることでV.Iの張替えを要求する。ステップS12ではプロセス162からの張替え完了報告を待つ。

【0041】

ステップS13では張替え完了フラグ123を立てる。張替え完了フラグ123を立てることにより、以後張替え要求トリガを発行させるイベントが発生した場合でも、二重に張替え要求が発行されること防ぐ。ステップS14ではV.I張替え処理のルーチンから復帰する場合は、休止していた親プロセス112の処理を再開し、そうでなければ通常の処理の実行する。

【0042】

ステップS15では親プロセス112の通常の処理において、プロセス162からのデータを受信したか否かをチェックする。この処理はRECV()をエミュレートした図10に示すEMU\_REDV()F5において実行される。ステップS15においてデータ受信が発生した場合はステップS16に進む。データ受信が発生していない場合はステップS17へ進む。

【0043】

ステップS16ではデータ既受信フラグ113を立てる。データ既受信フラグ113を立てることにより、子プロセス115にコピーすべきデータの受信が発生したこと親プロセス112は知る。

【0044】

ステップS17では親プロセス112の通常の処理において該プロセスが終了するか否かをチェックする。終了する場合はステップS18へ進み該プロセスは終了する。終了しない場合は再度ステップS5へ進み、これまでのステップを繰り返す。

【0045】

以上がVI張替えの際の親プロセス112の動作である。

【0046】

次に、図5に示すフローチャートによって子プロセス115の処理の流れを説明する。

【0047】

ステップS21ではEMU\_FORK()F1によって子プロセス115が生成されて処理が開始される。

【0048】

ステップS22では子プロセス115は自身におけるVI張替え要求トリガの発生の有無をチェックする。ステップS22においてトリガT3またはトリガT4が発生していればステップS23に進む。該トリガが発生していないければステップS26に進む。

【0049】

ステップS23では子プロセス115は張替え要求フラグC119を立て、親プロセス112に対して子プロセス115において張替え要求トリガが発生したことを報告する。親プロセス112は張替え要求フラグC119をスレッド114で監視して該トリガの発生を知る。

【0050】

ステップS24では子プロセス115は親プロセス112からのデータコピー要

求の有無をチェックする。該チェックはコピー要求監視スレッド141によってコピー発生フラグ140を監視することで行われる。ステップS24においてデータコピー要求がある場合（コピー発生フラグ140が1の場合）は、子プロセス115が実行している通常の処理を一旦休止し、ステップS25に進む。ステップS24においてデータコピー要求が無い場合（コピー発生フラグ140が0の場合）はステップS26に進む。

#### 【0051】

ステップS25では子プロセス115は親プロセス112のアドレス空間が子プロセス115へコピーされる。

#### 【0052】

ステップS26では子プロセス115は親プロセス112からのVI張替え要求の有無をチェックする。該チェックはスレッド116によって張替え要求フラグP118を監視することで行われる。ステップS26において張替え要求がある場合（張替え要求フラグP118が1の場合）は、VI張替え処理のルーチンであるステップS25に進む。ステップS26において張替え要求が無い場合（張替え要求フラグP118が0の場合）はステップS29に進む。

#### 【0053】

ステップS27では子プロセス115はクライアントアプリケーション161上のプロセス162からの接続要求を聴取する。

#### 【0054】

ステップS28では子プロセス115はプロセス162からの接続要求に対して接続許可を出す。

#### 【0055】

ステップS29ではVI張替え処理のルーチンから復帰する場合は、休止していた子プロセス115の処理を再開し、そうでなければ通常の処理の実行する。

#### 【0056】

ステップS30では子プロセス115の通常の処理において該プロセスが終了するか否かをチェックする。終了する場合はステップS31へ進み該プロセスは終了する。終了しない場合は再度ステップS22へ進み、これまでのステップを繰

り返す。

【0057】

以上がVI張替えの際の子プロセス115の動作である。

【0058】

最後に、図6に示すフローチャートでプロセス162の処理の流れを説明する。

【0059】

ステップS41で開始されたプロセス162はステップS42において親プロセス112に対してVIの接続要求を行う。

【0060】

プロセス162はステップS43では接続が許可されるのを待つ。

【0061】

ステップS44では親プロセス112からのデータ転送中断要求の有無をチェックする。該チェックは中断要求監視スレッド164によって中断要求フラグ163を監視することで行われる。ステップS44において中断要求がある場合（中断要求フラグ163が1の場合）は、プロセス162が実行している通常の処理を一旦休止し、ステップS45へ進む。ステップS44において中断要求が無い場合（中断要求フラグ163が0の場合）はステップS52へ進む。

【0062】

ステップS45ではデータ転送中断のための処理を行う。

【0063】

ステップS46ではデータ転送中断が完了したか否かをチェックする。終了していなければステップS45に戻る。終了していればステップS47に進む。

【0064】

ステップS47では親プロセス112に対してパケット通信によりデータ転送中断完了の報告を行う。

【0065】

ステップS48ではプロセス162は親プロセス112からのVI張替え要求の有無をチェックする。該チェックはスレッド164によって張替え要求フラグ163を監視することで行われる。ステップS48において張替え要求がある場合

(張替え要求フラグ163が1の場合)は、プロセス162が実行している通常の処理を一旦休止し、VI張替え処理のルーチンであるステップS49へ進む。ステップS48において張替え要求が無い場合(張替え要求フラグ163が0の場合)はステップS52へ進む。

#### 【0066】

ステップS49ではプロセス162は子プロセス115に対してVIの接続要求を行う。次にステップS50では接続が許可されるのを待つ。

#### 【0067】

接続が許可されると、プロセス162はステップS51で親プロセス112に対してVIの張替えが完了したことを報告する。該報告は親プロセス112に対して張替え完了報告用パケットを送信し張替え完了フラグ123を立てることにより行う。

#### 【0068】

次にステップS52ではプロセス162はVI張替え処理のルーチンから復帰する場合は、休止していたプロセス162の処理を再開し、そうでなければ通常の処理の実行する。

#### 【0069】

ステップS53ではプロセス162の通常の処理において該プロセスが終了するか否かをチェックする。終了する場合はステップS54へ進み該プロセスは終了する。終了しない場合は再度ステップS44へ進み、これまでのステップを繰り返す。

#### 【0070】

以上がVI張替えの際のプロセス162の動作である。

#### 【0071】

以上で説明した親プロセス112および子プロセス115およびプロセス162におけるそれぞれの処理が全体として時系列でどのように動作するかを図9に示すタイムチャートを用いて説明する。

#### 【0072】

まず時点aにおいてクライアントからサーバ上の親プロセスに対して接続要求9

10を発行し、該親プロセスは時点Aでそれを受け取る。なお図中、処理名称の始めに表記した括弧内の字句は該クライアントに対する処理の対象がサーバ上の親プロセスであるか子プロセスであるかを示す。以下同様である。続いて該親プロセスは時点Bで該クライアントに対して接続許可911を発行し、該クライアントは時点bでそれを受け取る。時点Xにおいて該親プロセスはEMU\_FORK()918によって子プロセスを生成する。時点Cにおいて該親プロセスは該クライアントに対してデータ転送中断要求912を発行し、該クライアントは時点cでそれを受け取る。時点c以降該クライアントはデータ転送中断処理(図示せず)を行い、中断処理が完了した時点dにおいて該親プロセスに対して中断完了報告913を発行する。該親プロセスは時点Dで該中断完了報告913を受け取り、その後時点Yにおいて該子プロセスへの既受信データのコピー922を行う。続いて時点Eにおいて該親プロセスは該クライアントに対して張替え要求914を発行し、該クライアントは時点eにおいてそれを受け取る。時点fにおいて該クライアントは該子プロセスに対して接続要求915を発行し、該子プロセスは時点Fにおいてそれを受け取る。続いて時点Gにおいて該子プロセスは該クライアントに対して接続許可916を発行し、該クライアントは時点gにおいてそれを受け取る。最後に時点hにおいて該クライアントは該親プロセスに対して張替え完了報告917を発行し、該親プロセスは時点Hでそれを受け取り、一連のVI張替え処理が完了する。

### 【0073】

ここで該親プロセスは時点B以降データ受信可能状態であり、該子プロセスは時点G以降データ受信可能状態である。また時点b c間および時点h以降が該クライアントがデータ送信可能時間であり、時点d g間が送信中断時間である。もし該クライアントが時点bでデータの送信を開始した場合、時点B'において該親プロセスが該データを受信可能になる。しかし該親プロセスは中断完了報告913を受け取ることにより、該親プロセスは時点D以降データの受信が無いことを知る。したがってVIの張替えの前に該親プロセスがデータを受け取る可能性のある時間は時点B' D間であり、その間にデータを受け取った場合に該データを該子プロセスにコピーする。その結果、VI張替え完了後に該子プロセスが正し

く該親プロセスの通信を引き継ぐことが可能になる。

#### 【0074】

一方図8は、データ転送の中止を行わなかった場合のタイムチャートである。図8を用いて本発明の有効性を示す。

#### 【0075】

まず図9と同様に該親プロセスは時点B以降データ受信可能状態であり、該子プロセスは時点G以降データ受信可能状態である。また時点b c間および時点h以降が該クライアントがデータ送信可能時間であり、時点d g間が送信中断時間である。しかしながら該親プロセスは該クライアントが送信を中断し、それがサーバ側に反映される時点E'を知ることはできない。すなわち該親プロセスは該クライアントからのデータがいつまで送信されるかを知ることができない。したがって該親プロセスはどの時点で該子プロセスに対して既受信データのコピーを行えばよいかわからず、コピーが正しく行われる保証ができない。

#### 【0076】

これに対し、本実施態様では、先に図9で説明した通り、サーバ上の親プロセスはクライアント上のプロセスにデータ転送中止要求を出し、クライアント側から中止完了が返送されることによって、それまでに受信したデータを子プロセスにコピーすれば良いことが分かる。よってVI張替えを行った場合に正しく通信を継続することを保証される。

#### 【0077】

以上述べた実施形態においては、プロセス間の通信は共有メモリを用いて実行しているが、その他のPIPE等の手段を使用してもよい。またホスト間の通信はパケット通信によって実行しているが、その他の手段を用いても良い。さらに第1のホストと第2のホストは物理的に分離されている必要はなく、同一情報処理装置上で実行されているプロセスにおいてもこの発明が適用されることは自明である。

#### 【0078】

##### 【発明の効果】

上述のように、本発明によれば、ユーザ空間直接転送によりOSの介在なしに通

信を行っているクライアント側のプロセスとサーバ側のプロセス（親プロセス）との間の通信回線を、自発的にサーバ側の別のプロセス（子プロセス）との間の通信回線に張替えて通信を継続させ、UNIXのFORK()と同様の機能を実現することが出来る。

【図面の簡単な説明】

【図1】

本発明の一実施形態のサーバ・クライアントシステムのブロック図である。

【図2】

一般的なTCP/IPにおけるソケット通信の概念を示す概念図である。

【図3】

ユーザ空間直接転送で通信するサーバ・クライアントシステムシステムにおける通信の状態を示すブロック図である。

【図4】

上記実施態様における通信路を張替える際のサーバ側の親プロセスの処理を示すフローチャート。

【図5】

上記実施態様における通信路を張替える際のサーバ側の子プロセスの処理を示すフローチャート。

【図6】

上記実施態様における通信路を張替える際のクライアント側のプロセスの処理を示すフローチャート。

【図7】

上記実施態様にて通信路を張替えるトリガとなるイベントを示す図。

【図8】

実施態様と比較のための通信路を張替える際にサーバ側の親プロセスがクライアント側のプロセスに対してデータ転送の中止を要求しない場合の処理の進行を示すタイムチャート。

【図9】

実施態様の通信路を張替える際の処理の進行を示すタイムチャート。

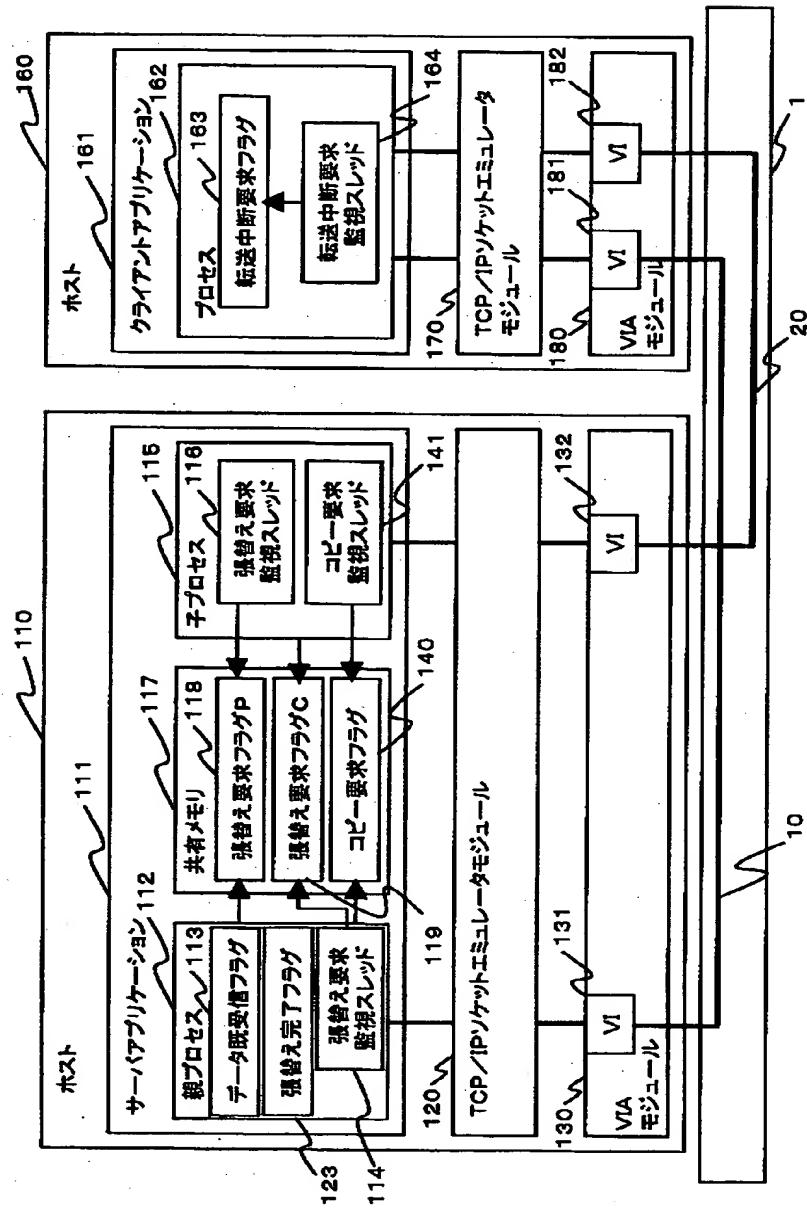
【符号の説明】

1…ネットワーク、10, 20…通信回線、110, 160…ホスト、111…サーバアプリケーション、112…親プロセス、116…子プロセス、120, 170…TCP/IPソケット通信エミュレータブロック、130, 180…VIAモジュール、131, 132, 181, 182…VI、161…クライアントアプリケーション、162…プロセス。

【書類名】 図面

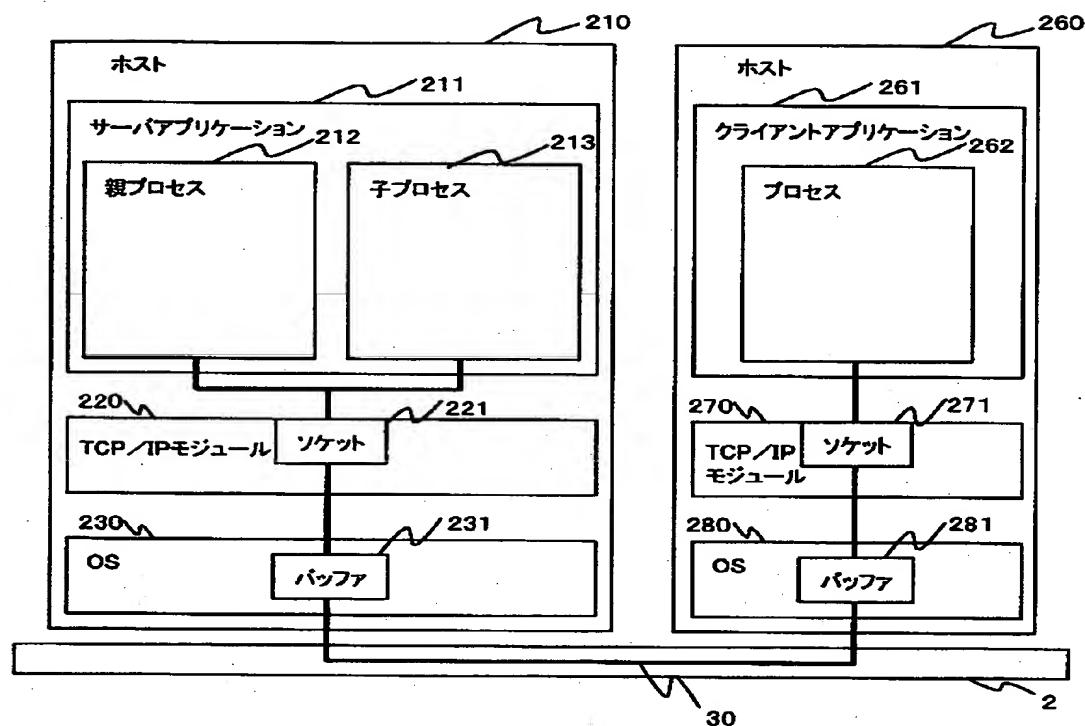
【図1】

1



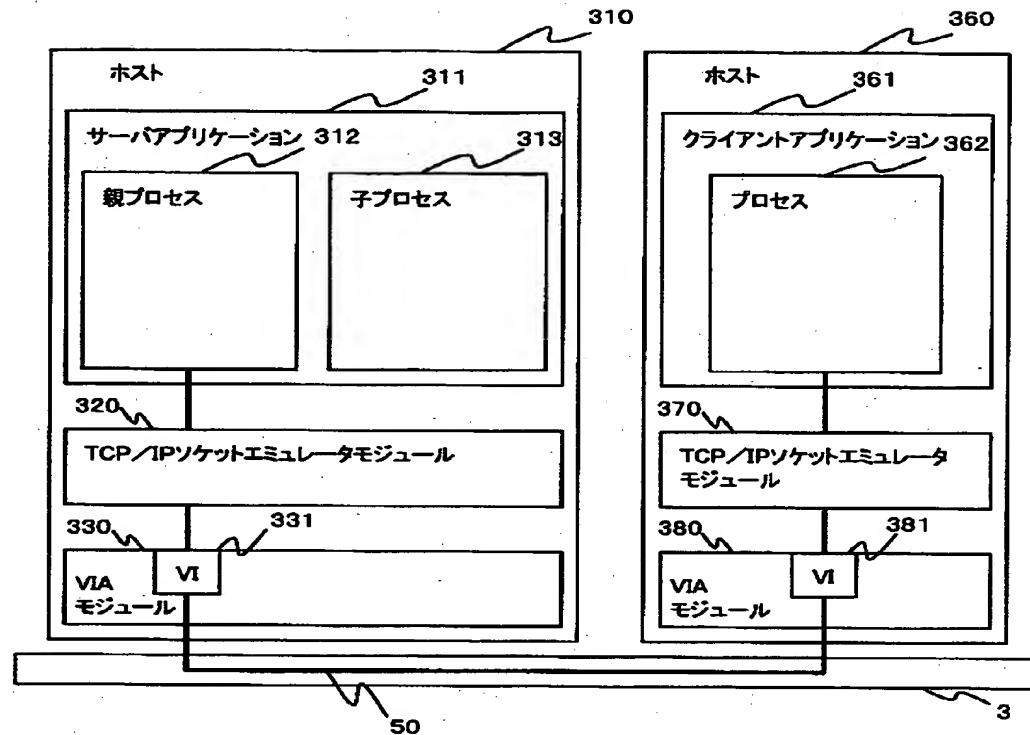
【図2】

図2



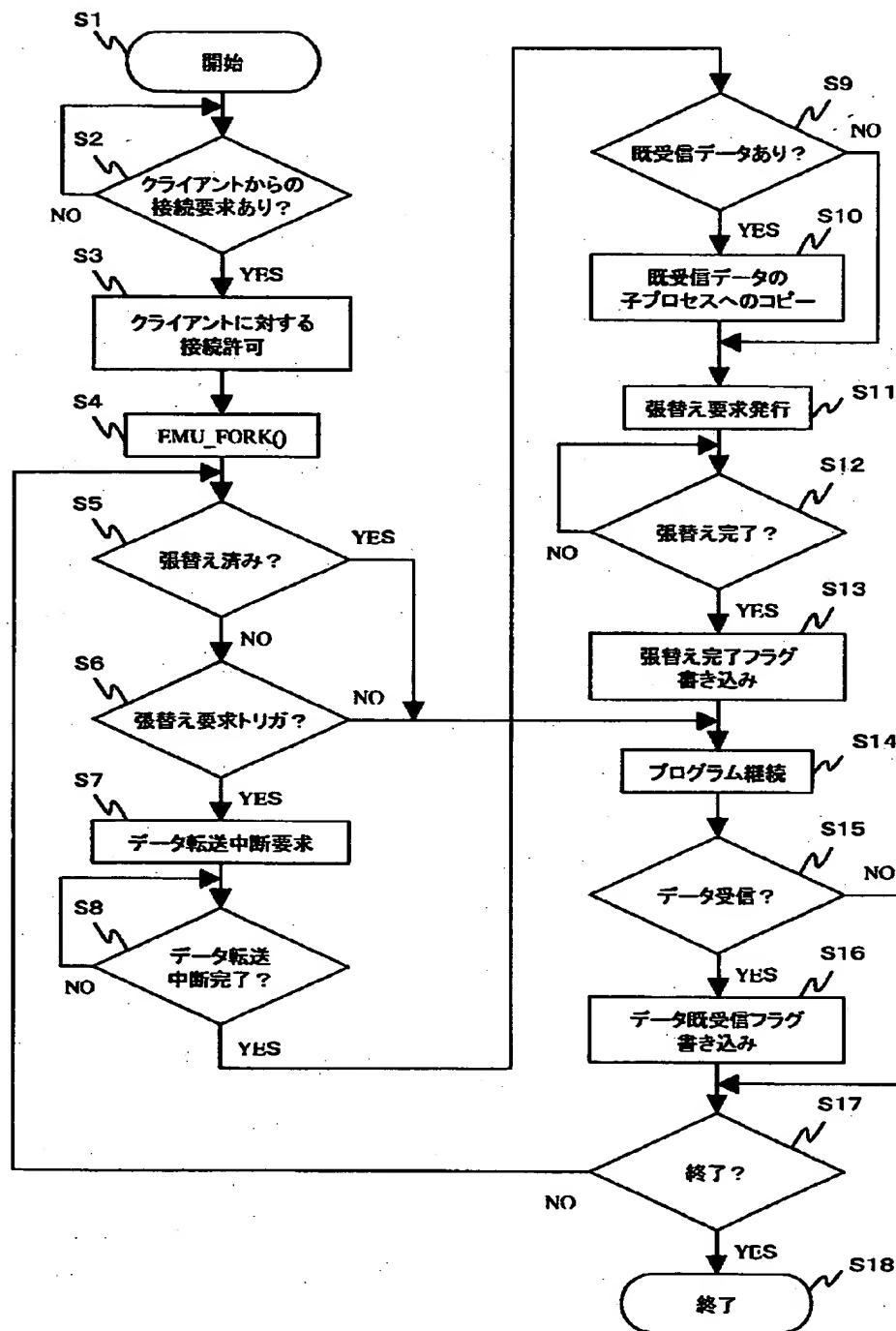
【図3】

図3



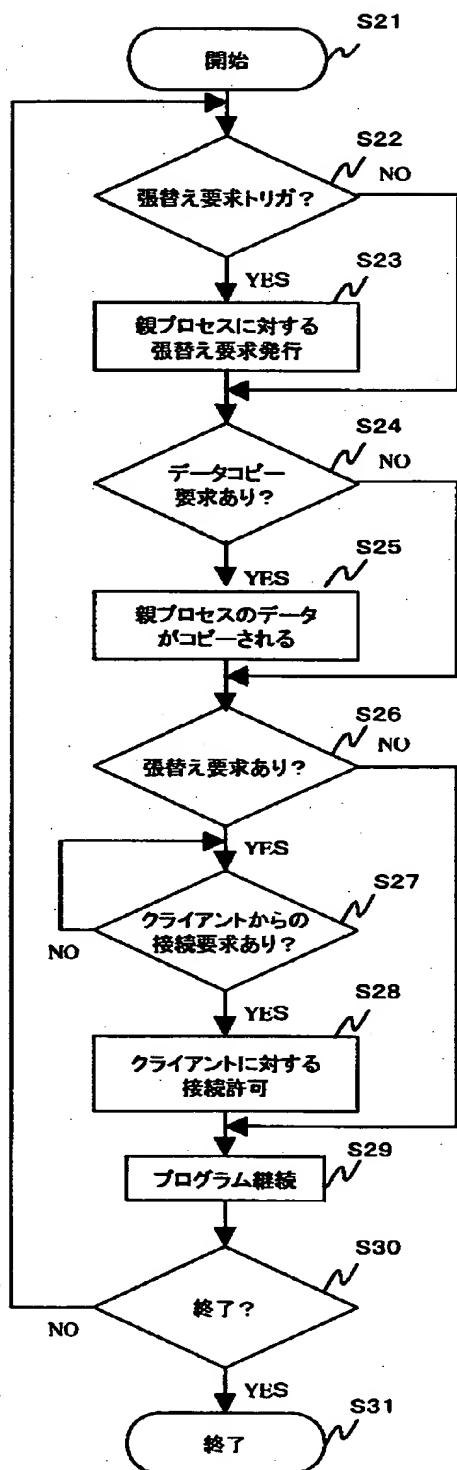
【図4】

図4



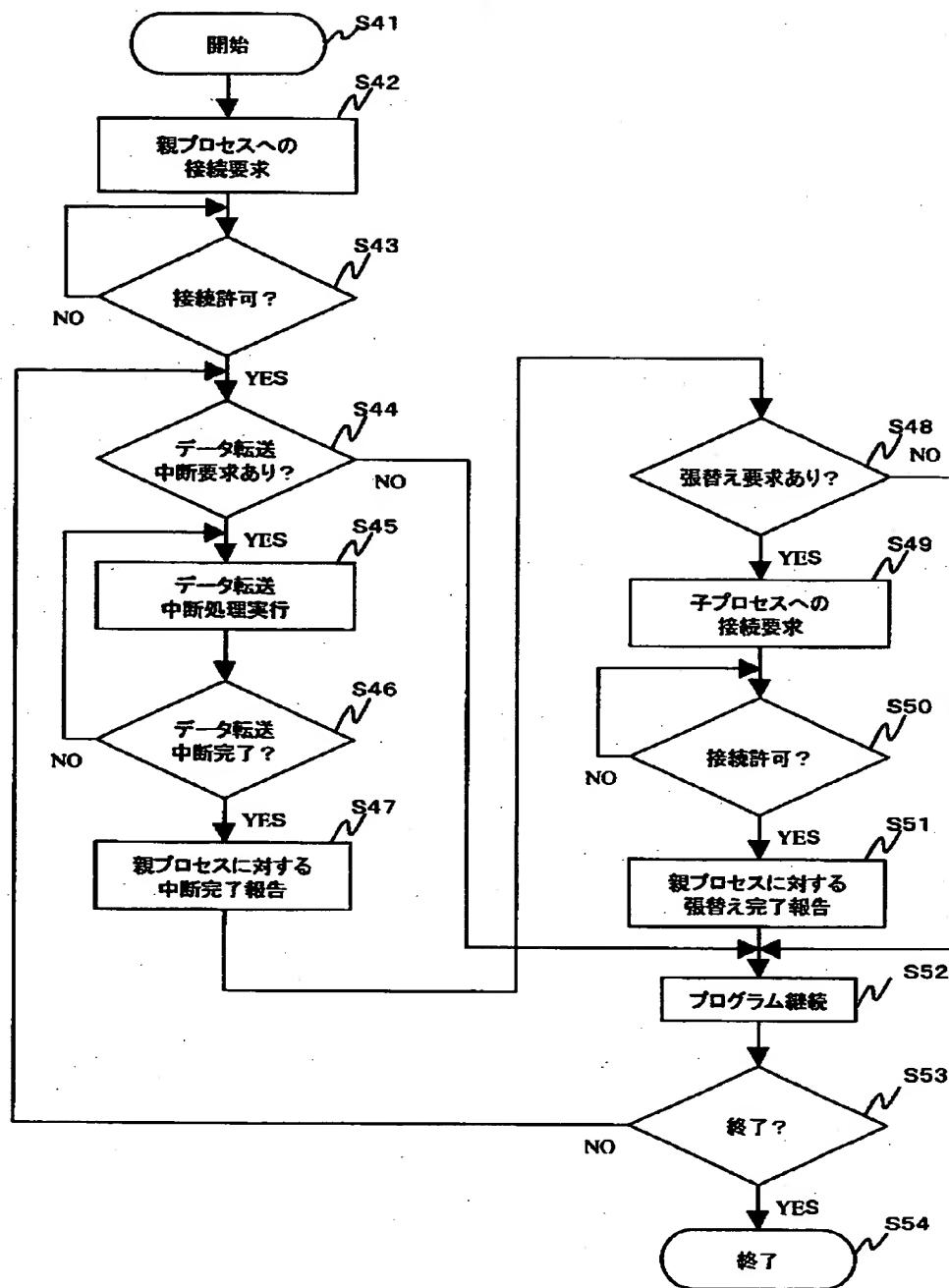
【図5】

図5



【図6】

図6



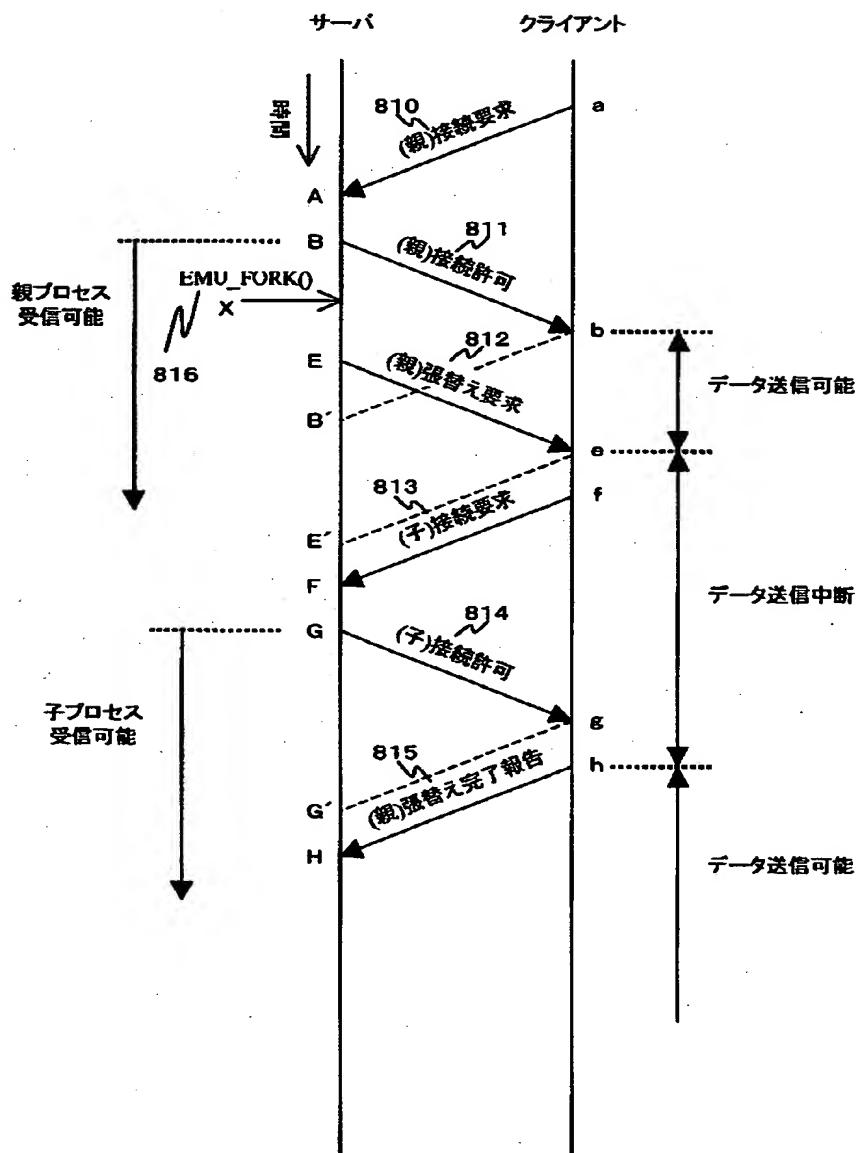
【図7】

図7

イベント	親プロセス	子プロセス
RECV	—	O
SEND	— T1	O T3
CLOSE	O T2	— T4
プロセス消滅	O	—

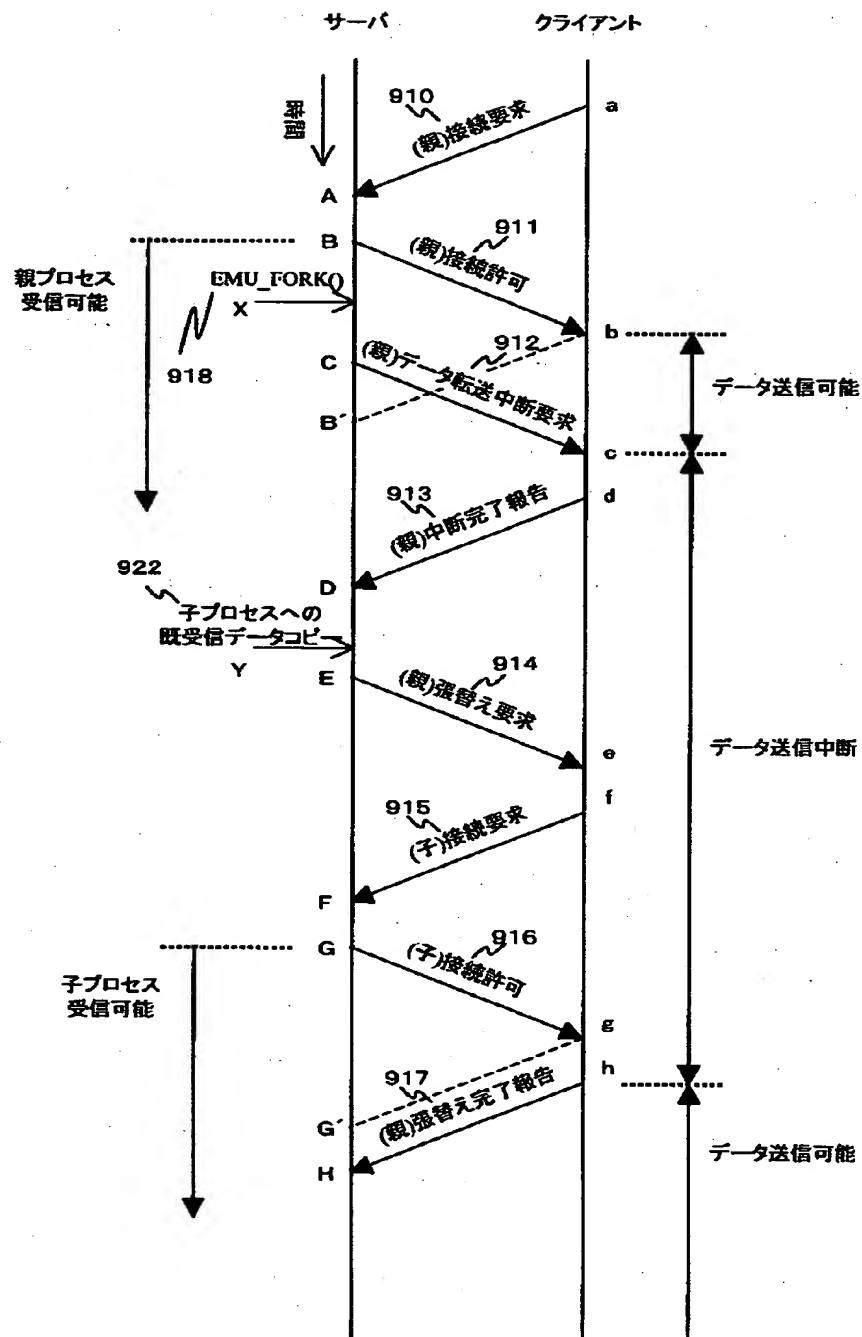
【図8】

図8



【図9】

図9



【書類名】 要約書

【要約】

【課題】 ユーザ空間直接転送によりOSの介在なしに通信を行っているクライアント側のプロセスと（親プロセス）サーバ側のプロセスとの間の通信回線を、自発的にサーバ側の別のプロセス（子プロセス）との間の通信回線に張替えて通信を継続させ、UNIXのFORKと同様の機能を実現する。

【解決手段】 サーバ側の第1のプロセスとクライアント側の第2のプロセスが第1の通信回線によって通信中に、第1のプロセスが第2のプロセスに対して通信の中止を要求し、第2のプロセスは第1のプロセスに中断完了の報告を行う機能と、第1のプロセスがプロセス生成機能により生成したサーバ側の第3のプロセスと第2のプロセスに対して互いに第2の通信回線の接続を確立するように要求する機能と、前記接続を確立するまでに第1のプロセスがすでに第2のプロセスからのデータを受信している場合は該データを第3のプロセスにコピーする機能とを持つことで第2の通信が正しく第1の通信を引き継ぐことが出来る。

【選択図】 図1

出願人履歴情報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日

[変更理由] 新規登録

住所 東京都千代田区神田駿河台4丁目6番地  
氏名 株式会社日立製作所